Text Classification and Sequence Labeling

Graham Neubig
Text Classification

- Given an input text $X$, predict an output label $y$

### Topic Classification

<table>
<thead>
<tr>
<th>I like peaches and pears</th>
<th>food</th>
<th>politics</th>
<th>music</th>
</tr>
</thead>
<tbody>
<tr>
<td>I like peaches and herb</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Language Identification

<table>
<thead>
<tr>
<th>I like peaches and pears</th>
<th>English</th>
<th>Japanese</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>桃と梨が好き</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Sentiment Analysis (sentence/document-level)

<table>
<thead>
<tr>
<th>I like peaches and pears</th>
<th>positive</th>
<th>neutral</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>I hate peaches and pears</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

... and many many more!
Sequence Labeling

- Given an input text $X$, predict an output label sequence $Y$ of equal length!

**Part of Speech Tagging**

<table>
<thead>
<tr>
<th>He</th>
<th>saw</th>
<th>two</th>
<th>birds</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRON</td>
<td>VERB</td>
<td>NUM</td>
<td>NOUN</td>
</tr>
</tbody>
</table>

**Lemmatization**

<table>
<thead>
<tr>
<th>He</th>
<th>saw</th>
<th>two</th>
<th>birds</th>
</tr>
</thead>
<tbody>
<tr>
<td>he</td>
<td>see</td>
<td>two</td>
<td>bird</td>
</tr>
</tbody>
</table>

**Morphological Tagging**

<table>
<thead>
<tr>
<th>He</th>
<th>saw</th>
<th>two</th>
<th>birds</th>
</tr>
</thead>
<tbody>
<tr>
<td>PronType=prs</td>
<td>Tense=past, VerbForm=fin</td>
<td>NumType=card</td>
<td>Number=plur</td>
</tr>
</tbody>
</table>

... and more!
Span Labeling

• Given an input text \( X \), predict an output spans and labels \( Y \).

Named Entity Recognition

Graham Neubig is teaching at Carnegie Mellon University

- **PER**
- **ORG**

Syntactic Chunking

Graham Neubig is teaching at Carnegie Mellon University

- **NP**
- **VP**
- **NP**

Semantic Role Labeling

Graham Neubig is teaching at Carnegie Mellon University

- **Actor**
- **Predicate**
- **Location**

... and more!
Span Labeling as Sequence Labeling

- Predict **Beginning**, **In**, and **Out** tags for each word in a span.

<table>
<thead>
<tr>
<th>Graham Neubig is teaching at Carnegie Mellon University</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PER</strong></td>
</tr>
<tr>
<td><strong>ORG</strong></td>
</tr>
</tbody>
</table>

```
Graham Neubig is teaching at Carnegie Mellon University

B-PER  I-PER  O  O  O  B-ORG  I-ORG  I-ORG
```
Text Segmentation

• Given an input text $X$, split it into segmented text $Y$.

Tokenization

A well-conceived "thought exercise."
A well-conceived "thought exercise."

Word Segmentation

外国语人参政権

foreign people voting rights

Morphological Segmentation

Köpekler

Köpek ler

dog Number=Plural

Köpekle r

dog_paddle Tense=Aorist

• Rule-based, or span labeling models
Modeling for Sequence Labeling/Classification
How do we Make Predictions?

- Given an input text $X$
- Extract features $H$
- Predict labels $Y$

Text Classification

I like peaches

Feature Extractor

Predict

positive

Sequence Labeling

I like peaches

Feature Extractor

Predict

Predict

Predict

PRON VERB NOUN
A Simple Extractor:
Bag of Words (BOW)

I
lookup
like
lookup
peaches
lookup
=
Predict
Label Probs
A Simple Predictor: Linear Transform + Softmax

\[ p = \text{softmax}( W \ast h + b ) \]

- Softmax converts arbitrary scores into probabilities

\[ p_i = \frac{e^{s_i}}{\sum_j e^{s_j}} \]

\[ s = \begin{pmatrix} -3.2 \\ -2.9 \\ 1.0 \\ 2.2 \\ 0.6 \\ \cdots \end{pmatrix} \quad \rightarrow \quad p = \begin{pmatrix} 0.002 \\ 0.003 \\ 0.329 \\ 0.444 \\ 0.090 \\ \cdots \end{pmatrix} \]
Problem: Language is not a Bag of Words!

I don’t love pears

There’s nothing I don’t love about pears
Better Featurizers

- Bag of n-grams
- Syntax-based features (e.g. subject-object pairs)
- Neural networks
  - Recurrent neural networks
  - Convolutional networks
- Self attention
What is a Neural Net?: Computation Graphs
“Neural” Nets

Original Motivation: Neurons in the Brain

Current Conception: Computation Graphs

$f(x_1, x_2, x_3) = \sum x_i$

$f(M, v) = Mv$

$f(U, V) = UV$

$f(u) = u^T$

$f(u, v) = u \cdot v$
expression:

\[ x \]

graph:

A **node** is a \{tensor, matrix, vector, scalar\} value
An **edge** represents a function argument.

A **node** with an incoming **edge** is a **function** of that edge’s tail node.

A **node** knows how to compute its value and the value of its derivative w.r.t each argument (edge) times a derivative of an arbitrary input $\frac{\partial F}{\partial f(u)}$.

\[
\frac{\partial f(u)}{\partial u} \frac{\partial F}{\partial f(u)} = \left( \frac{\partial F}{\partial f(u)} \right)^T
\]
Functions can be nullary, unary, binary, … $n$-ary. Often they are unary or binary.

$$f(U, V) = UV$$

$$f(u) = u^T$$
expression:

$$x^T A x$$

graph:

Computation graphs are generally directed and acyclic
expression:
\[ x^T A x \]

graph:
expression:
\[ x^\top A x + b \cdot x + c \]

graph:

\[ f(x_1, x_2, x_3) = \sum_i x_i \]

\[ f(M, v) = Mv \]

\[ f(U, V) = UV \]

\[ f(u) = u^\top \]

\[ f(u, v) = u \cdot v \]
expression:
\[ y = x^T A x + b \cdot x + c \]

graph:

\[ f(x_1, x_2, x_3) = \sum_i x_i \]
\[ f(M, v) = Mv \]
\[ f(U, V) = UV \]
\[ f(u) = u^T \]
\[ f(u, v) = u \cdot v \]

variable names are just labelings of nodes.
Algorithms (1)

- Graph construction

- Forward propagation
  - In topological order, compute the value of the node given its inputs
Forward Propagation

graph:

\[ f(x_1, x_2, x_3) = \sum_i x_i \]

\[ f(M, v) = Mv \]

\[ f(U, V) = UV \]

\[ f(u) = u^\top \]

\[ f(u, v) = u \cdot v \]
Forward Propagation

**graph:**

\[ f(x_1, x_2, x_3) = \sum_{i} x_i \]

\[ f(M, v) = Mv \]

\[ f(U, V) = UV \]

\[ f(u) = u^T \]

\[ f(u, v) = u \cdot v \]
Forward Propagation

graph:

\[ f(x_1, x_2, x_3) = \sum_{i} x_i \]

\[ f(U, V) = UV \]

\[ f(M, v) = Mv \]

\[ f(u) = u^T \]

\[ f(u, v) = u \cdot v \]
Forward Propagation

graph:

\[ f(x_1, x_2, x_3) = \sum_i x_i \]

\[ f(U, V) = UV \]

\[ f(M, v) = Mv \]

\[ f(u) = u^\top \]

\[ f(u, v) = u \cdot v \]
Forward Propagation

graph:

\[ f(x_1, x_2, x_3) = \sum_i x_i \]

\[ f(\mathbf{u}, \mathbf{v}) = \mathbf{u} \cdot \mathbf{v} \]

\[ f(\mathbf{M}, \mathbf{v}) = \mathbf{Mv} \]

\[ f(\mathbf{U}, \mathbf{V}) = \mathbf{UV} \]

\[ f(\mathbf{u}) = \mathbf{u}^\top \]

\[ \mathbf{x}^\top \mathbf{A} \]

\[ \mathbf{x}^\top \]

\[ \mathbf{A} \]

\[ \mathbf{b} \]

\[ \mathbf{c} \]
Forward Propagation

graph:

\[ f(x_1, x_2, x_3) = \sum_i x_i \]
\[ f(M, v) = Mv \]
\[ f(U, V) = UV \]
\[ f(u) = u^T \]
\[ f(u, v) = u \cdot v \]
Forward Propagation

\[ f(x_1, x_2, x_3) = \sum_i x_i \]

Graph:

\[ f(M, v) = Mv \]
\[ f(U, V) = UV \]
\[ f(u) = u^T \]
\[ f(u, v) = u \cdot v \]
Forward Propagation

graph:

\[
f(x_1, x_2, x_3) = \sum x_i
\]

\[
x^\top Ax + b \cdot x + c
\]

\[
f(M, v) = Mv
\]

\[
f(U, V) = UV
\]

\[
f(u) = u^\top
\]

\[
f(u, v) = u \cdot v
\]
Algorithms (2)

• **Back-propagation:**
  • Process examples in reverse topological order
  • Calculate the derivatives of the parameters with respect to the final value (This is usually a “loss function”, a value we want to minimize)

• **Parameter update:**
  • Move the parameters in the direction of this derivative
    \[ W \leftarrow \alpha \times \text{dl/dW} \]
Back Propagation

graph:

\[ f(x_1, x_2, x_3) = \sum x_i \]

\[ f(M, v) = Mv \]

\[ f(U, V) = UV \]

\[ f(u) = u^\top \]

\[ f(u, v) = u \cdot v \]
Neural Network Frameworks

Examples in this class
Basic Process in (Dynamic) Neural Network Frameworks

• Create a model

• For each example

  • *create a graph* that represents the computation you want

  • *calculate the result* of that computation

  • if training, perform *back propagation and update*
Recurrent Neural Networks
Long-distance Dependencies in Language

• Agreement in number, gender, etc.
  
  He does not have very much confidence in himself.  
  She does not have very much confidence in herself.

• Selectional preference

  The reign has lasted as long as the life of the queen.  
  The rain has lasted as long as the life of the clouds.
Recurrent Neural Networks
(Elman 1990)

• Tools to “remember” information
Unrolling in Time

• What does featurizing a sequence look like?
Representing Sentences

- Text Classification
- Conditioned Generation
- Retrieval
Representing Words

- Sequence Labeling
- Language Modeling
- Calculating Representations for Parsing, etc.
Training RNNs

I like these pears

RNN → RNN → RNN → RNN → RNN

predict → predict → predict → predict → predict

prediction 1 → prediction 2 → prediction 3 → prediction 4

loss 1 → loss 2 → loss 3 → loss 4

label 1 → label 2 → label 3 → label 4

sum → total loss
RNN Training

- The unrolled graph is a well-formed (DAG) computation graph—we can run backprop

- Parameters are tied across time, derivatives are aggregated across all time steps

- This is historically called “backpropagation through time” (BPTT)
Parameter Tying

Parameters are shared! Derivatives are accumulated.

I like these pears.
Bi-RNNs

- A simple extension, run the RNN in both directions

I like these pears
Multilingual Labeling/Classification
Data and Models
Language Identification

**LTI Language Identification Corpus**
http://www.cs.cmu.edu/~ralf/langid.html
- Benchmark on 1152 languages from a variety of free sources

**langid.py**
https://github.com/saffsd/langid.py
- Off-the-shelf language ID system for 90+ languages

**Automatic Language Identification in Texts: A Survey**
Text Classification

- Very broad field, many different datasets

<table>
<thead>
<tr>
<th>MLDoc: A Corpus for Multilingual Document Classification in Eight Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://github.com/facebookresearch/MLDoc">https://github.com/facebookresearch/MLDoc</a></td>
</tr>
<tr>
<td>• Topic classification, eight languages</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAWS-X: Paraphrase Adversaries from Word Scrambling, Cross-lingual Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://github.com/google-research-datasets/paws/tree/master/pawsx">https://github.com/google-research-datasets/paws/tree/master/pawsx</a></td>
</tr>
<tr>
<td>• Paraphrase detection (sentence pair classification)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cross-lingual Natural Language Inference (XNLI) corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://cims.nyu.edu/~sbowman/xnli/">https://cims.nyu.edu/~sbowman/xnli/</a></td>
</tr>
<tr>
<td>• Textual entailment prediction (sentence pair classification)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cross-lingual Sentiment Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available from: <a href="https://github.com/ccsasuke/adan">https://github.com/ccsasuke/adan</a></td>
</tr>
<tr>
<td>• Chinese-English cross-lingual sentiment dataset</td>
</tr>
</tbody>
</table>
Part of Speech/ Morphological Tagging

- Part of universal dependencies treebank
  [https://universaldependencies.org/](https://universaldependencies.org/)
- Contains parts of speech and morphological features for 90 languages
- Standardized "Universal POS" and "Universal Morphology" tag sets make things consistent
- Several pre-trained models on these datasets:
  - Udify: [https://github.com/Hyperparticle/udify](https://github.com/Hyperparticle/udify)
  - Stanza: [https://stanfordnlp.github.io/stanza/](https://stanfordnlp.github.io/stanza/)

Named Entity Recognition

"Gold Standard"

CoNLL 2002/2003 Language Independent Named Entity Recognition
- English, German, Spanish, Dutch human annotated data

"Silver Standard"

WikiAnn Entity Recognition/Linking in 282 Languages
https://www.aclweb.org/anthology/P17-1178/
Available from: https://github.com/google-research/xtreme
- Data automatically extracted from Wikipedia using inter-page links
Composite Benchmarks

• Benchmarks that aggregate many different sequence labeling/classification tasks

**XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization**

https://github.com/google-research/xtreme

• 10 different tasks, 40 different languages

**XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation**

https://microsoft.github.io/XGLUE/

• 11 tasks over 19 languages (including generation)
Discussion Exercise
Universal Dependencies Comparison

• Download data from the Universal Dependencies Treebank for one language you know, and one language that you do not know and is very different

  https://universaldependencies.org/

• Look at the part-of-speech and morphological tags, and think about what you can tell from them

• For example: is the word order different between the languages? does one language have richer morphology? what else can you see?
Thank You!